AMENDMENTS TO THE CLAIMS

Claims 1-10 (Cancelled)

11. (Currently Amended) A method for context-based serialization of messages in a server system having a message queue and an execution system which performs parallel processing of messages, comprising the steps of:

retrieving a message from the message queue;

~~determining~~ identifying a transaction identifier of the message;

searching a state table for the transaction identifier;

creating an entry for the message in the state table with the transaction identifier and a state of active [[if]] upon the state table ~~does~~ not containing the identified transaction identifier ~~found in the step of determining~~; and

putting the message into an execution queue.

12. (Withdrawn) A method for automatic context-based serialization of messages in a server system having an execution system for parallel processing of messages, comprising the steps of:

retrieving a message from a message queue;

analyzing the message to find a transaction identifier;

searching a state table for the transaction identifier of the message; and

putting the message into a serialization queue if the state table includes the transaction identifier with an active state.

13. (Withdrawn) The method according to claim 12, further comprising the steps of:

checking states of messages in the state table at predetermined times; and

if a message with the transaction identifier has been cleared in the state table, putting a message from the serialization queue having the transaction identifier into the execution queue.

14. (Withdrawn) The method according to claim 13, further comprising the steps of:

recording, in the state table, a time stamp of a message arriving in the message queue; and

putting the message into the execution queue in a sequence according the time stamp.

15. (Withdrawn) The method according to claim 12, further comprising the steps of:

assigning a message ID to the transaction identifier in the state table; and

retrieving the message from the serialization queue using the message ID.

16. (New) A messaging system for context-based serialization of messages in a parallel execution environment, comprising

a serialization processor for

receiving an incoming message from a message client,

extracting a transaction identifier from the message,

checking a state table for the transaction identifier, putting the message into a serialization queue if the transaction identifier is found with an active status in the state table, and putting the message into an execution queue once the transaction identifier has been cleared in the state table.

17. (New) The messaging system according to claim 16, wherein

upon the transaction identifier with the state active being found in the state table, the serialization processor

enters a status of queued in the state table with a status of queued and

puts the message into the serialization queue.

18. (New) The messaging system according to claim 16, wherein

the serialization processor

creates a time stamp entry in the state table, and the time stamp entry defines an arrival time of a message, and

puts messages into the execution queue in a sequence according to time stamps of the messages.

19. (New) The messaging system according to claim 16, wherein

the serialization processor creates a message ID entry in the state table for use by the serialization processor to retrieve messages from the serialization queue.

20. (New) A server system, comprising:

a messaging server communicating with a messaging client;

a queue manager for managing a message queue; and an execution system for parallel processing of messages;

wherein the server system includes:

a serialization queue for storing messages to be queued;

an execution queue for storing messages to be executed;

a state table providing state information; and

a serialization processor for taking messages from the message queue responsive to the state information and updating the state table accordingly.

21. (New) The server system according to claim 20, wherein

the state table comprises a transaction identifier and present state of a message,

and the present state is selected as active or queued.

22. (New) The server system according to claim 21, wherein

the state table further comprises a time stamp recording the arrival time of the message.

23. (New) The server system according to claim 22, wherein

the state table further comprises a message identifier for the message for retrieval of the message from the serialization queue.

24. (New) The server system according to claim 20, wherein

the serialization processor

receives an incoming message from the messaging client via the message queue,

reads a transaction identifier included in the message, checks the state table for the transaction identifier,

puts the message into a serialization queue upon the transaction identifier already stored in the state table with the state active, and

puts the message into an execution queue upon the entry in the state table with the transaction identifier has been cleared.

25. (New) The server system according to claim 24, wherein

the serialization processor makes an entry in the state table with a status of queued upon the transaction identifier already being stored in the state table with the state active.